

# A New Dual-Differential Full-Adder Design for CED-based Fault-Tolerant Circuits

Mouna Karmani, Noura Ben Hadjyoussef, Belgacem Hamdi, Mohsen Machhout

*Faculty of Sciences of Monastir, Electronics and MicroElectronics Laboratory (LEME),  
Monastir 5019, Tunisia.*

karmani.mouna@fsm.rnu.tn, benhadjyoussef.noura@fsm.rnu.tn

**Abstract** — *In this work, we present a new dual-differential full-adder bit-slice design that can be used for implementing circuits with Concurrent Error Detection (CED) capability. The proposed design is suitable to realize applications needing diagnostics and maintenance in the field. The proposed design used a mixed DPL/PTL logic style design, requires only 24 transistors, and provides dual-differential outputs. To prove the proposed design efficiency, the circuit is designed and simulated using the standard 32 nm technology node. The proposed scheme can be used to ameliorate the circuit fault detection, fault-masking, and then fault tolerance capability. By using the proposed full-adder bit-slice, we have implemented and simulated a 4-bit ripple carry adder design realized using a mixed CMOS/DPL/PTL logic style design. The simulation results prove the acceptable electrical behavior of the implemented 4-bit ripple carries adder design.*

**Keywords** — *full-adder; dual-differential; CED; 32nm technology node;*

## I. INTRODUCTION

Due to the very large scale integration densities, integrated circuits design has been and still is an actual challenge to ensure the needed level of accuracy and reliability. In fact, the CMOS technology continues to be the dominant hardware design solution because of its several advantages such as low power and cost, high operating clock and density, performance, and, especially, manufacturing designer's experience [1]. However, with the continuous scaling of CMOS devices, the exponential increase in power density has elevated the temperature of the on-chip system [2]. In fact, the leakage currents responsible for power dissipation increased significantly in advanced CMOS technologies, and designers' effort has been concentrated in order to develop design techniques for power saving [3]-[6]. Thus, for digital circuits, the power density increasing is a real problem that leads to very high temperatures and can affect the correct system behavior. Therefore, VLSI circuits have become more susceptible to transient faults, and reliability concerns are pronounced more strongly for all computing systems [7]. Consequently, elaborating efficient and reliable designs is an important challenge for embedded systems with safety-critical constraints. In fact, the complexity of deep sub-micron VLSI

circuits causes faults to occur during the normal circuit operation, and many of these faults (such as transient ones) cannot be detected using offline testing techniques. However, many critical faults might be detected in real-time during the normal circuit operation [8]. In fact, transient faults are becoming an increasingly serious concern which makes concurrent checking more attractive thanks to its capability of detecting transient faults occurring in a circuit while operating normally. The Concurrent Error Detection (CED) capability is the property of verifying the circuit's delivered results during its normal operation. The CED capability offers the possibility of self-diagnosis and self-correction for designs used in critical application domains needing great reliability levels [9]. In this work, we have firstly proposed a new dual-differential full adder bit-slice design using a mixed Double Pass-transistor Logic (DPL)/PTL Pass-transistor Logic (PTL) style design. The proposed full adder implementation can be used for schemes using the dual duplication code. To demonstrate the proposed design efficiency, the circuit is implemented using the full-custom 32 nm technology generation. The circuit SPICE simulations extracted from the layout and including parasitic, prove that the proposed circuit has a conformed electrical behavior. In order to evaluate the efficiency of our proposed design, faults are voluntarily injected into the circuit's layout. According to the dual-differential full adder simulation results, the proposed architecture shows a trade-off between area overhead and CED capability. Finally, by using the proposed full adder bit-slice, we have proposed and implemented a 4-bit ripple carry adder design. The paper is organized as follows. The CED-based fault-tolerant systems are presented in section 2. Section 3 and 4 describe the proposed design and provide simulation results. Finally, conclusions are summarized in section 5.

## II. CED-BASED FAULT-TOLERANT SYSTEMS

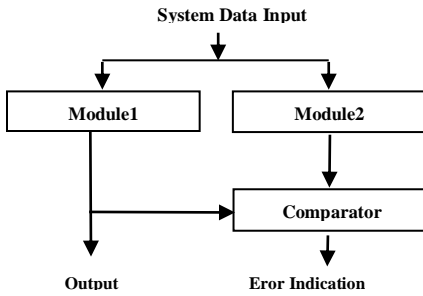
The incapability to exactly control the fabrication process at deep-submicron technology nodes causes process variations [10]. So, at small-feature technologies with low voltage levels, devices are more susceptible to permanent and transient faults. As a consequence, the fault-tolerance property could be used for tolerating faults within safety-critical system designs. Therefore, building CED fault-tolerant devices is crucial for safety-critical applications to assure the correctness of the



computed result even with the presence of permanent and/or transient faults [11, 12]. In the following sections, we will enumerate some CED techniques that can be used in order to make the circuit more tolerant to transient and permanent faults.

**A. The hardware redundancy CED technique**

For most safety-critical applications, the corresponding level of safety needs to detect any single fault which comes out while operating normally. To assure this online fault detection property, we may use the CED approaches. In fact, hardware redundancy is the most primary CED technique. Figure 1 illustrates a hardware redundancy example, which is the Dual Modular Redundancy (DMR). For this form of hardware redundancy, two identical hardware copies (module 1 and module 2) are used concurrently to execute the same data computation, and the two obtained results are compared, and then any discrepancy is considered an error [13].

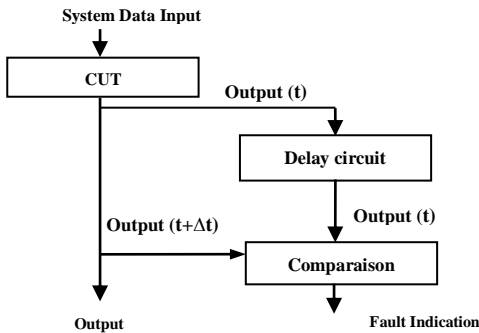


**Fig. 1. A duplex system**

The major problem with this technique is its impact on physical weight, size, power consumption, and cost. The Dual Modular Redundancy method requires 100% area overhead to achieve the self-checking capability for hardware units, which is not an ideal solution for constrained resource designs.

**B. The temporal redundancy CED technique**

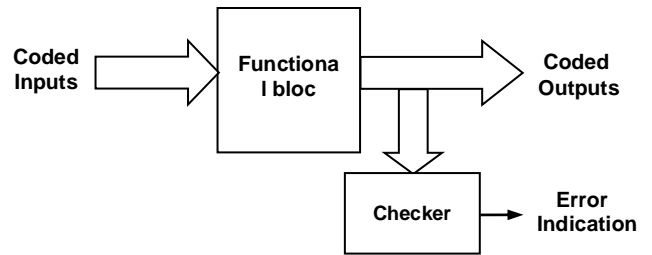
The time redundancy CED technique can be used to cut down the amount of area overhead induced by hardware redundancy techniques but uses extra time. However, for applications with real-time constraints, additional time may be more catastrophic than extra hardware [14]. The temporal redundancy concept is illustrated in Figure 2.



**Fig. 2. The temporal redundancy technique**

**C. The dual duplicated CED technique**

The dual duplicated scheme is a well-known self-checking circuit design methodology. By using this design methodology, a circuit is divided into its constituent functional blocks, where each of these blocks is implemented according to the structures illustrated in Figure 3. This technique implements functional blocks generating outputs according to an error detecting code. The concurrent error detection is ensured using a checker added to monitor the coded outputs [15]. Self-checking circuits elaborate online testing during the normal operation mode of a system without applying any extra tests [16]. Also, being Totally Self-checking (TSC), the considered circuit can immediately detect the error of an electronic system to avoid data damage or the malfunction of a function circuit. The basic structure of TSC circuits is given in Fig. 1.



**Fig. 3. The totally self-checking circuit basic structure**

Hence, the TSC capability can enhance the reliability of an electronic system, i.e., using TSC may reduce the damage of the electronic system to the lowest as we wish. A circuit is TSC if both the functional block and the checker are TSC. A functional bloc is TSC if it is fault secure and self-testing. The checker compares two input words that should normally be complementary and generate dual functions and determines whether the circuit output is valid or not [17]. In fact, designing efficient adders with fault tolerance capability is of great importance [18]. Therefore, the proposed dual-differential full-adder design is suitable for implementing self-checking functional blocks by providing dual-differential outputs to be used by a Code checker circuit in order to determine whether the circuit’s output is a valid code-word or not [19, 20].

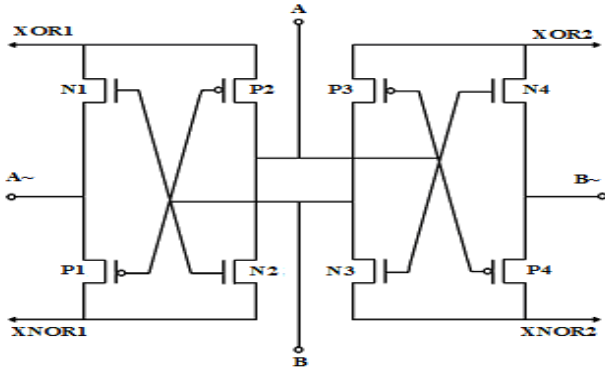
**III. THE PROPOSED DUAL DIFFERENTIAL SELF-CHECKING ADDER**

A Full Adder is a three-input two-output circuit, where  $A_i$  and  $B_i$  are the inputs to be summed,  $C_i$  is the carry input bit which derives from the calculations of the previous sum operation. The circuit outputs are  $S_i$ , the result of the current sum operation, and  $C_{out}$ , the carry output value.

$$\begin{cases} S_i = A_i \oplus B_i \oplus C_i \\ C_{outi} = (A_i \oplus B_i) \cdot A_i + (A_i \oplus B_i) \cdot C_i \end{cases}$$

In fact, exclusive OR (denoted XOR) and exclusive NOR (denoted XNOR) gates are well known for their important role in self-checking adders and arithmetic units implemented

with the dual duplication technique. The XOR/XNOR functions can be implemented using different logic style designs such as pass-transistor logic, double pass transistor logic, transmission gates, and static CMOS logic [11-12]. The proposed dual differential full adder design is performed using our dual-differential XOR/XNOR gate proposed in previous work [9] and illustrated by Figure 4.



**Fig.4. The XOR-XNOR implementation with duplicated output computation**

As illustrated in Figure 5, The XOR-XNOR gate has dual inputs ( $A_i, A_i\sim$ ) and ( $B_i, B_i\sim$ ) and generates the ( $XOR1, XNOR1$ ) and ( $XOR2, XNOR2$ ) dual duplicated outputs. The XOR-XNOR circuit design is carried out using eight MOS transistors. In this case, the fault-tolerance property is ensured by using a dual duplicated CED technique by providing duplicated dual outputs by using two-path output computations. The first path gives the first dual outputs ( $XOR1, XNOR1$ ), while the second one provides the second dual outputs ( $XOR2, XNOR2$ ). As indicated, the XOR-XNOR outputs are carried out using two different paths. The  $XOR1$  and  $XNOR1$  outputs, described by equations (1) and

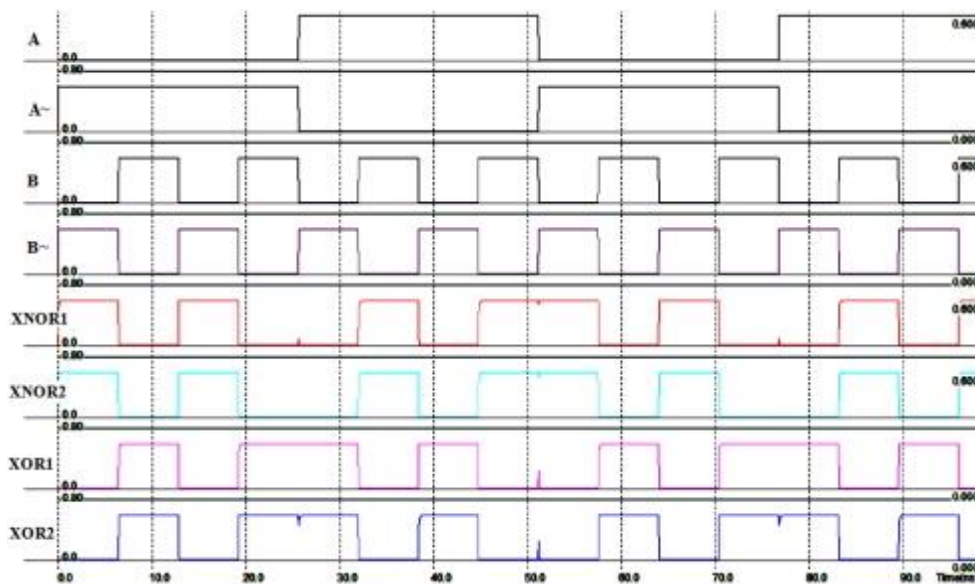
(2), are obtained From the first path.

$$\begin{cases} XOR1.= A\oplus B=A\sim B + A(B\sim) = P_{i1} & (1) \\ XNOR1.=(A\oplus B)\sim = AB + A\sim(B\sim) = P_{i1\sim} & (2) \end{cases}$$

For this first path, the dual inputs are ( $A, A\sim$ ) and ( $B, B\sim$ ) such that  $B\sim$  is obtained from the input  $B$ . Therefore, the current design will be insensitive to different types of errors affecting the input  $B\sim$ . The  $XOR2$  and  $XNOR2$  outputs, described by equations (3) and (4), are obtained using a second path. For the second path, the dual inputs are ( $A, A\sim$ ) and ( $B, B\sim$ ) such that  $A\sim$  is obtained from the input  $A$ . Therefore, the circuit will be insensitive to any errors affecting the input  $A\sim$ .

$$\begin{cases} XOR2.= A\oplus B=(A\sim)B + AB\sim = P_{i2} & (3) \\ XNOR2.=(A\oplus B)\sim = AB + (A\sim)B\sim = P_{i2\sim} & (4) \end{cases}$$

Therefore, if an error occurs, it will affect only one path, and the error can be revealed by verifying the complementarity between each ( $XOR, XNOR$ ) output. The XOR-XNOR circuit is implemented in a full-custom 32 nm technology node [21]. The circuit SPICE simulations, extracted from the layout, are performed to prove that the circuit has a conformed electrical behavior. SPICE simulation of the circuit without any fault is illustrated in Fig. 5. From the simulation results, we can remark that the obtained outputs are complementary. In addition, we must note that the current design doesn't generate degraded signals but produces strong 0's and 1's. This is very important for deep-submicron technologies having low voltage levels and small noise margins.

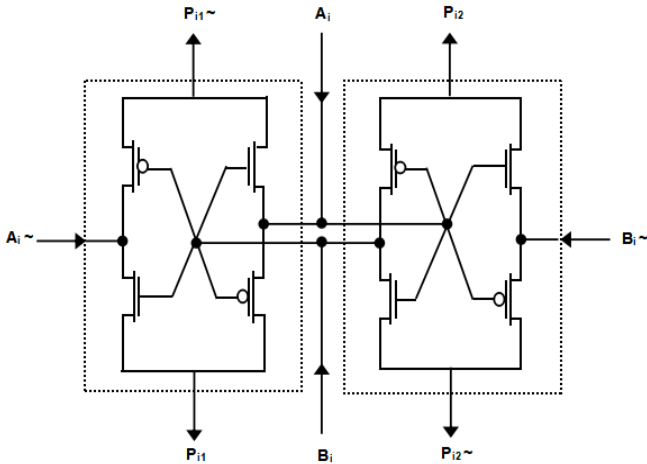


**Fig. 5. The XOR-XNOR SPICE simulation in 32nm technology**

As specified in the previous section, the XOR-XNOR structure is the basic brick of the implemented full adder circuit. In the following, we will explain all the building blocks of the proposed dual-differential full-adder design.

**A. The dual differential XOR-XNOR function**

In the following figure, we implement  $(P_{i1}, P_{i1}\sim)$  and  $(P_{i2}, P_{i2}\sim)$  using the xor-xnor gate such that  $P_{i1}=P_{i2}=A_i\oplus B_i$  and  $P_{i1}\sim=P_{i2}\sim=(A_i\oplus B_i)\sim$ .

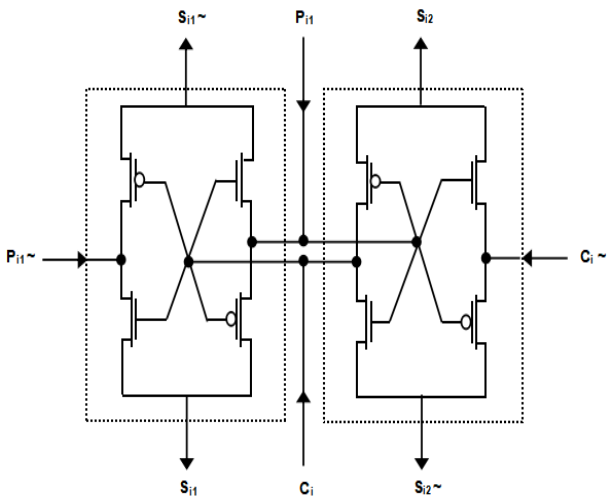


**Fig.6. The dual differential XOR/XNOR implementation**

$(P_{i1}, P_{i1}\sim)$  or  $(P_{i2}, P_{i2}\sim)$  signals and the dual carry inputs  $(C_i, C_i\sim)$  will be the inputs of the second differential XOR/XNOR gate that will generate the dual sum function outputs as illustrated in Figure 6.

**B. The dual differential sum function**

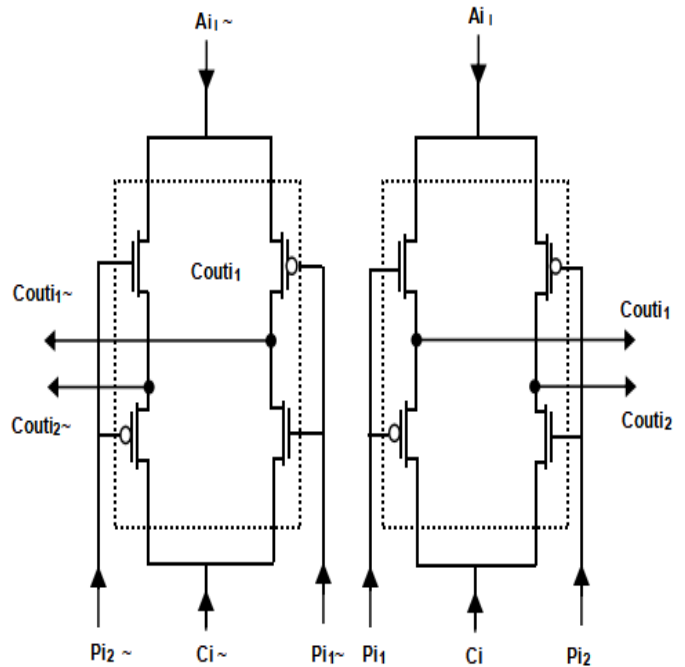
As shown in Figure 7, the second dual differential XOR/XNOR performs  $(S_{i1}, S_{i1}\sim)$  and  $(S_{i2}, S_{i2}\sim)$  the dual differential sum function outputs such that  $S_{i1}=S_{i2}=P_{i1}\oplus C_i$  and  $S_{i1}\sim=S_{i2}\sim=(P_{i1}\oplus C_i)\sim$ .



**Fig.7. The dual differential SUM function implementation**

**C. THE Dual differential CARRY GATE DESIGN**

The static CMOS differential carry gate design produces only  $C_{outi}$  and  $C_{outi}\sim$  differential outputs and used 20 transistors [21]. In this work, the dual differential carry outputs are performed in pass transistor technology, and only 8 transistors are required to implement the dual differential carry outputs  $(C_{outi1}, C_{outi2}, C_{outi1}\sim,$  and  $C_{outi2}\sim)$  which provide a significant gain in hardware overhead compared to CMOS counterparts. In fact, in literature, the Pass-Transistor Logic style design is suitable as fewer transistors are needed to perform important logic functions; it is also faster than conventional CMOS. In addition, the proposed dual differential carry gate doesn't generate degraded signals because it uses the dual differential inputs  $(P_{i1}, P_{i1}\sim)$  and  $(P_{i2}, P_{i2}\sim)$  such that each input is performed using a different path. Figure 8 illustrates the implemented dual differential carry gate.



**Fig.8. The dual differential Carry function implementation**

The described dual carry gate is used to produce the dual differential carry outputs of the proposed full adder design. Figure 9 gives the scheme of the self-checking full adder. This dual-differential implementation needs only 24 transistors. Commonly designers used CMOS inverters to restore degraded signals, which is noted the case for our proposed bit-slice design.

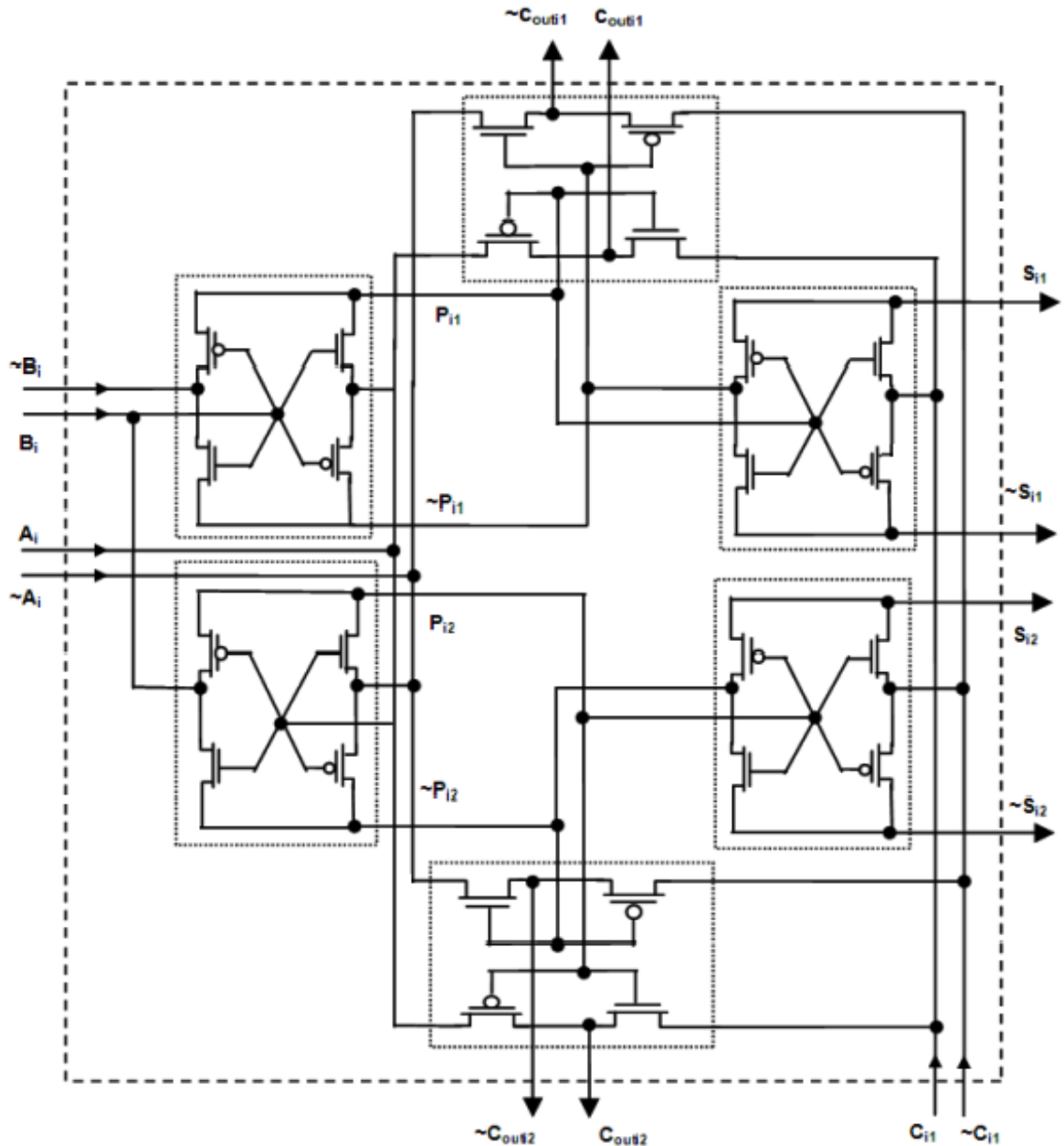


Fig.9. The proposed dual differential full adder (24 transistors)

#### IV. IMPLEMENTATION AND SIMULATION RESULTS

The proposed dual differential full adder design is implemented in full-custom 32 nm technology. The dual differential full adder layout is given in Figure 10. The circuit SPICE Simulation in the absence of any fault is illustrated by Figure 11. The simulation results prove that the circuit has an acceptable electrical behavior.

As illustrated by Figure 11, the dual differential adder circuit generates dual differential outputs which are the dual differential XOR/XNOR outputs: ( $P_{11}$ ,  $P_{11}\sim$ ) and ( $P_{12}$ ,  $P_{12}\sim$ ), the dual differential sum signals outputs ( $S_{11}$ ,  $S_{11}\sim$ ) and ( $S_{12}$ ,  $S_{12}\sim$ ) and the dual differential carry outputs ( $C_{out1}$ ,  $C_{out1}\sim$ ) and ( $C_{out2}$ ,  $C_{out2}\sim$ ).

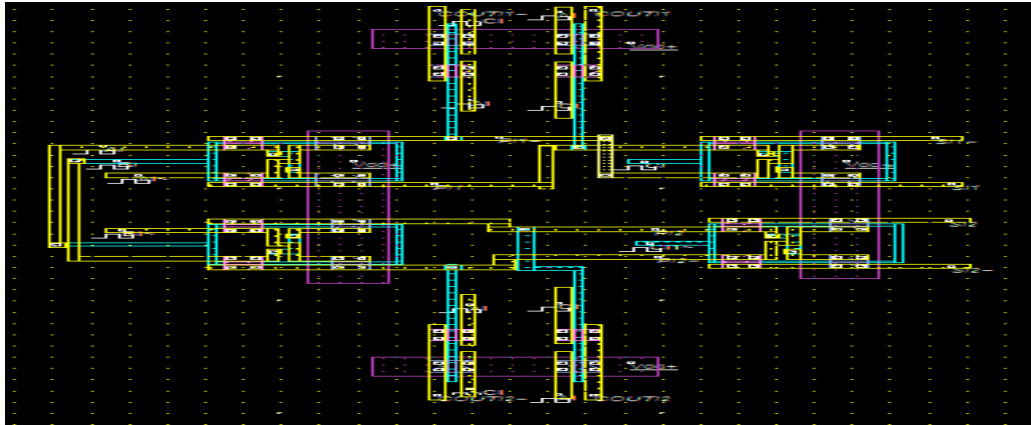


Fig.10. The layout of the dual differential full adder in full-custom 32 nm process technology

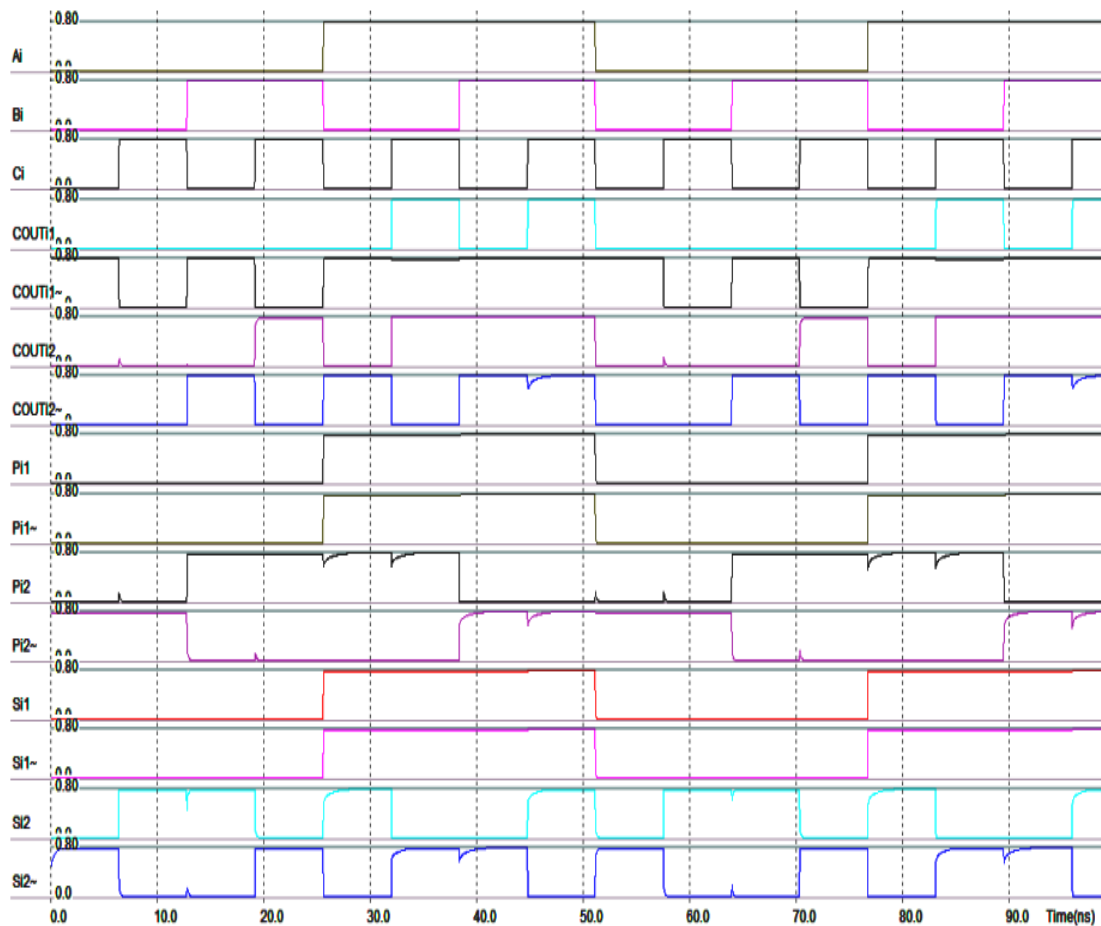


Fig. 11. The proposed dual-differential full-adder SPICE simulation results in 32nm technology

In order to verify the circuit's CED capability, we simulate the dual-differential full-adder bit-slice design in the presence of faults. Faults can be voluntarily and manually injected into the circuit's physical layout. In this case, the fault injected in

the primary input is  $(A_i = A_i\sim)$ , which can be the permanent or transient fault. The considered SPICE simulations are shown in Figure 12, while Table 1 summarizes the dual-differential full-adder internal states and outputs with  $(A_i=A_i\sim)$ .



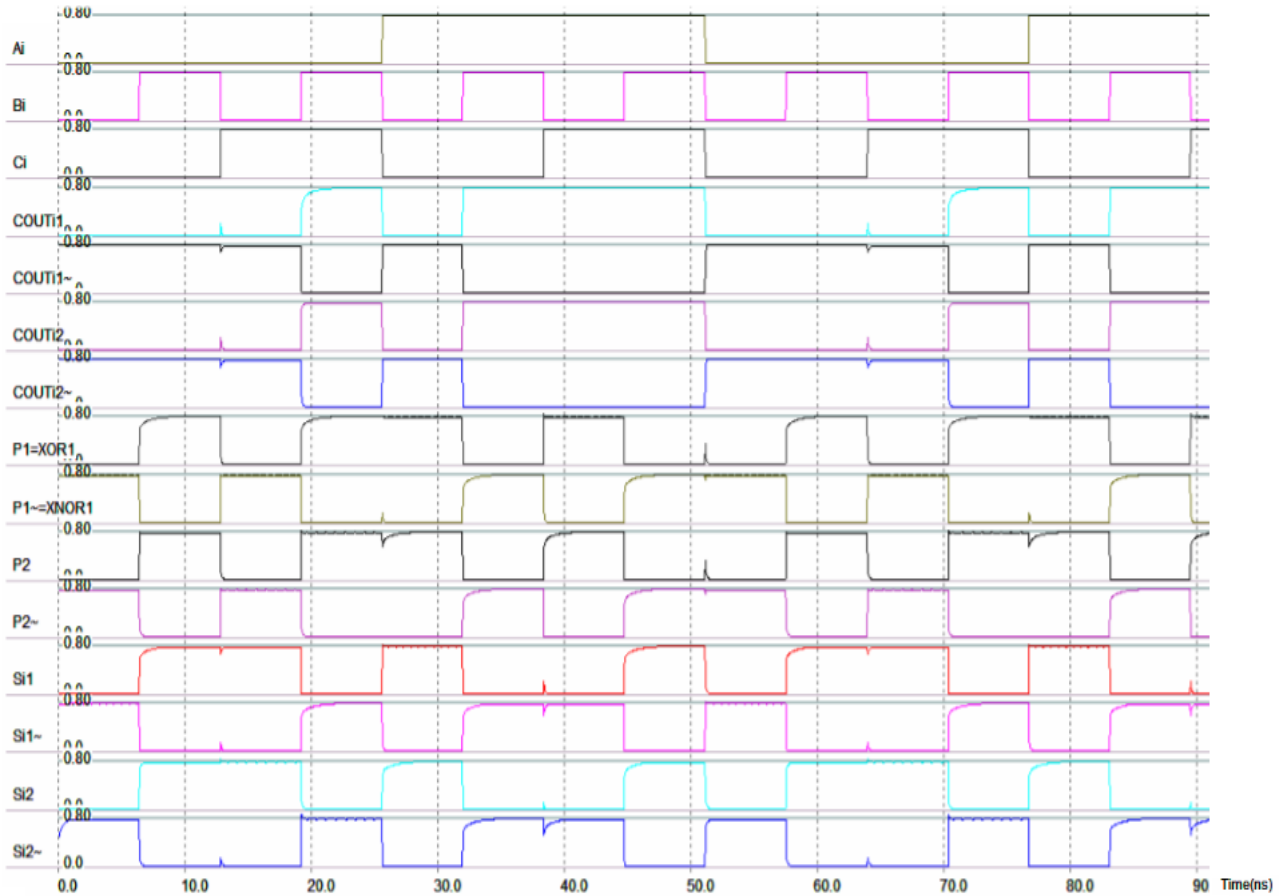


Fig. 12. The proposed dual-differential full-adder SPICE simulation results with an injection of primary fault ( $A_i=A_i\sim$ )

TABLE I. THE DUAL-DIFFERENTIAL FULL-ADDER INTERNAL STATES AND OUTPUTS RESPONSE WITH ( $A_i=A_i\sim$ )

Inputs				Internal states		Outputs				Conclusion
$A_i$	$B_i$	$B_i\sim$	$C_i$	$P_{i1}$ $P_{i1}\sim$	$P_{i2}$ $P_{i2}\sim$	$S_{i1}$ $S_{i1}\sim$	$S_{i2}$ $S_{i2}\sim$	$C_{out1}$ $C_{out1}\sim$	$C_{out2}$ $C_{out2}\sim$	
0	0	0	1	0 0	0 1	0 0	0 1	0 1	0 0	fault detected and corrected
0	1	0	1	0 0	0 1	0 0	1 0	0 0	0 0	fault detected
0	0	1	0	0 0	1 0	0 0	1 0	0 1	0 1	fault detected and corrected
0	0	1	0	0 0	1 0	0 0	0 1	0 0	1 0	fault detected and corrected
1	1	0	1	1 1	1 0	1 1	1 0	0 1	0 1	fault detected and corrected
1	1	0	1	1 1	1 0	1 1	0 1	1 1	1 0	fault detected and corrected
1	1	1	0	1 1	0 1	1 1	0 1	0 1	1 1	fault detected and corrected
1	1	1	0	1 1	0 1	1 1	1 0	1 1	1 1	fault detected

The primary fault injection ( $A_i=A_i\sim$ ) results in a non-valid code word and produces no complementary outputs, as shown in table 1. In fact, when the fault is injected, the dual differential outputs ( $S_{i1}/S_{i1}\sim$  and  $S_{i2}/S_{i2}\sim$ ) and carry ( $C_{outi1}/C_{outi1}\sim$  and  $C_{outi2}/C_{outi2}\sim$ ) don't remain complementary, which indicate a non-valid code. As seen in the table above, the fault is detected for all cases and corrected in the majority of cases.

The ripple carries adder is the simplest and the most common type of multiple-bit adder circuit [23]. However, it is not preferred in many applications because of its high latency [24]. We have chosen this adder deliberately in order to study the effect of carrying propagation on the output signals

Using the proposed dual-differential full-adder design, we have realized a dual-differential 4-bit ripple carry adder. The 4-bit adder block diagram is illustrated in Figure 13. The 4-bit dual differential full adder of Figure 13 is implemented in full-custom 32nm CMOS technology at a 0.8V power supply [21]. The circuit SPICE simulations, extracted from the layout, are illustrated by Figures 15 and 16, while the layout is given by Figure 14.

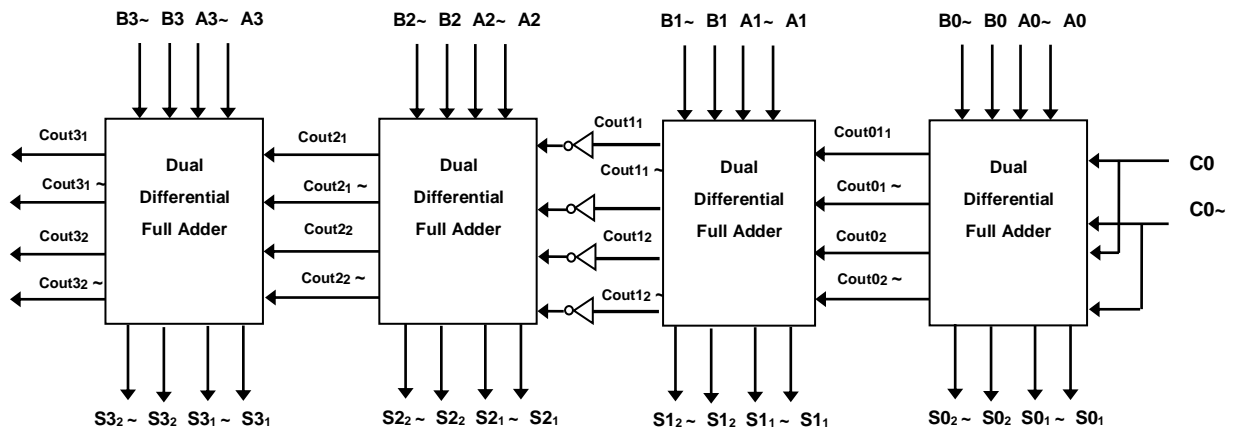


Fig.13. The 4-bit dual differential adder block diagram

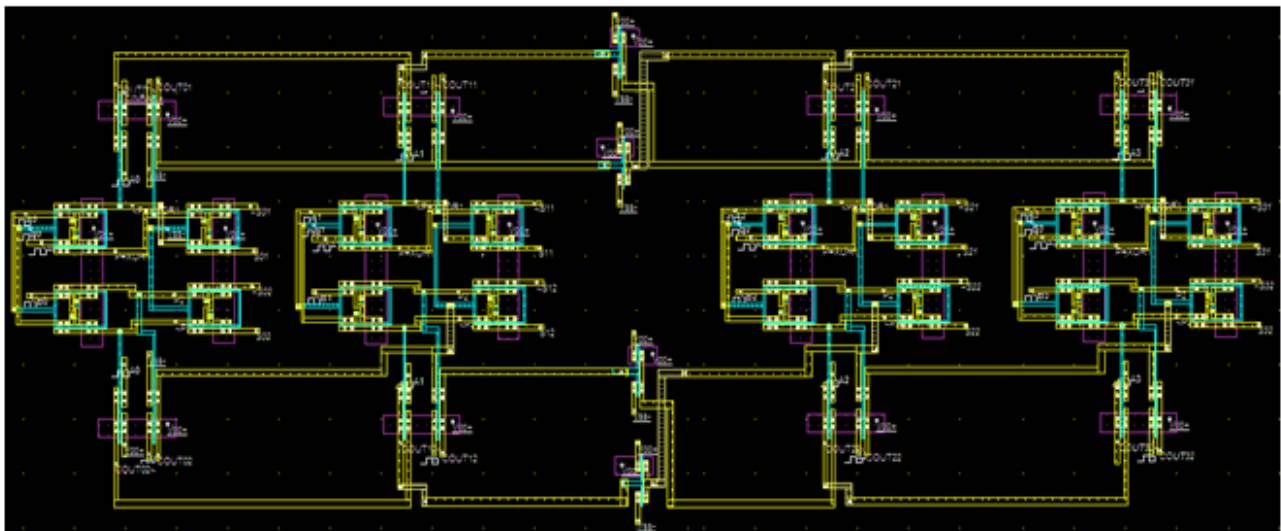
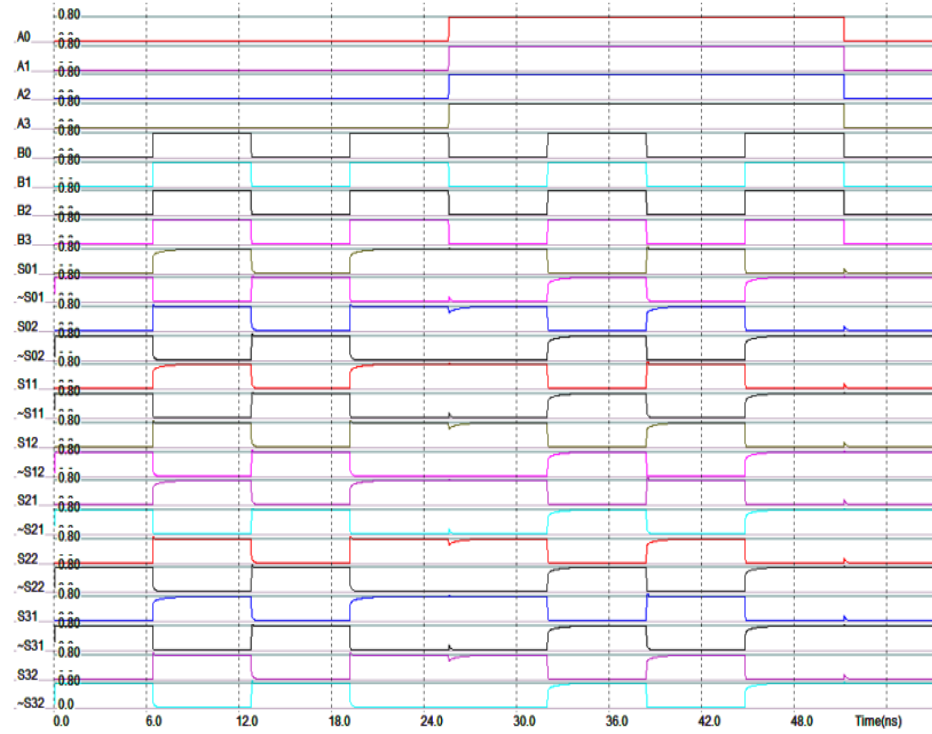


Fig.14. The 4-bit RCA layout design

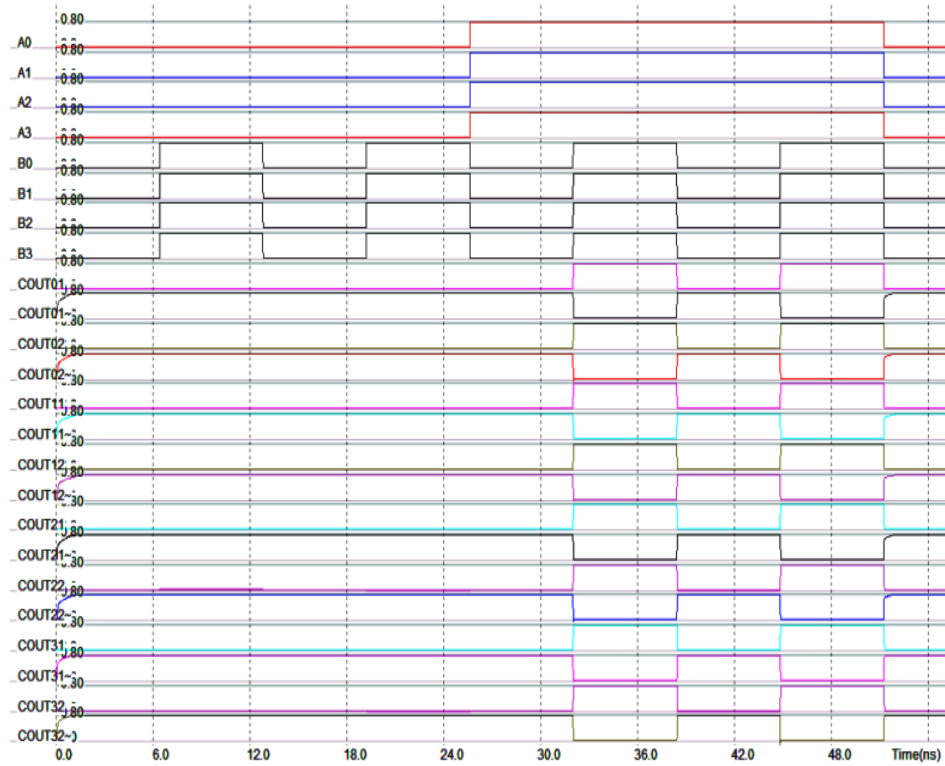
The described RCA design is implemented in full-custom 32 nm technology. The circuit SPICE Simulation in the absence of any fault is illustrated in Figures 15 and 16. The

simulations results demonstrate that the implemented circuit has an acceptable electrical behavior.





**Fig. 15.** The 4-bit ripple carry adder design SPICE simulation results in 32nm technology: the dual differential sum outputs



**Fig. 16.** The 4-bit ripple carry adder design SPICE simulation results in 32nm technology: All dual differential carry outputs

## V. CONCLUSION

The difficulty to exactly control the fabrication process at deep-submicron technology nodes causes process variation and makes the elaborated devices more vulnerable to transient and permanent faults. As a consequence, fault-tolerant techniques should be used for tolerating faults within safety-critical system designs and ensuring the correctness of the computed results even in the presence of permanent and transient failures. A self-checking circuit is carried out using functional blocks performed based on the dual complementation of each basic function; it has dual inputs and generates dual outputs. In this paper, we proposed a new dual-differential full-adder bit-slice implementation using the dual-differential duplication code. This scheme is based on a mixed DPL/PTL logic style design and requires only 24 transistors. By using the proposed full-adder bit-slice, we have implemented and simulated a 4-bit ripple carry adder design realized using a mixed CMOS/DPL/PTL logic style design. Simulation results demonstrate the acceptable electrical behavior of the dual-differential full-adder bit-slice and the 4-bit ripple carry adder design implemented both using the 32 nm technology node.

## REFERENCES

- [1] Karmani M, Benhadjoussef N, Hamdi B, Machhout M, The DFA/DFT-based hacking techniques and countermeasures: A case study of the 32-bit AES encryption crypto-core., *IET Comput. Digit. Tech.*, (2021) 160–170.
- [2] J.Hua, Y. Peng, Y. Xu†, K. Cao, J. Jia, Makespan Minimization for Multiprocessor Real-Time Systems under Thermal and Timing Constraints, *Journal of Circuits, Systems and Computers* 28, (2019).
- [3] K. Roy, S. Mukhopadhyay, and H. M.-Meimand, Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits, *Proceedings of the IEEE*, 91(2) (2003) 302-327.
- [4] S. Yang, W. Wolf, N. Vijaykrishnan, Y. Xie and W. Wang, Accurate Stacking Effect Macro-modeling of Leakage Power in Sub-100nm Circuits, *Proc. Int. Conference on VLSI Design*, (2005) 165-170.
- [5] Z. Cheng, M. Johnson, L. Wei, and K. Roy, Estimation of Standby Leakage Power in CMOS Circuits Considering Accurate Modeling of Transistor Stacks, *Proc. Int. Symposium Low Power Electronics and Design*, (1998) 239-244.
- [6] R. X. Gu and M. I. Elmasry, Power Distribution Analysis and Optimization of Deep Submicron CMOS Digital Circuit, *IEEE J. Solid-State Circuits*, 31(5) (1996) 707-713.
- [7] D. Zhu and H. Aydin, Reliability effects of process and thread redundancy on-chip multiprocessors, in *Proc. 36th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Networks*, (2006) 212–213.
- [8] Biswal, P.K., Biswas, S, A Binary Decision Diagram Approach to On-line Testing of Asynchronous Circuits with Dynamic and Static Elements, *J Electron Test* 35, (2019) 715–727.
- [9] M. Karmani, C. Khedhiri, B. Hamdi, K.L. MAN, e.g., LIM, C. LEI, A Concurrent error detection and correction based fault-tolerant XOR-XNOR circuit for highly reliable applications, *IAENG Transactions on Electrical Engineering*, 1 (2013) 56-69.
- [10] S. R. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, VARIUS: A model of process variation and resulting timing errors for microarchitectures, in *IEEE Transactions on Semiconductor Manufacturing*, (2008).
- [11] E. F. Hitt and D. Mulcar, *Fault-Tolerant Avionic*, CRC Press LL, (2001).
- [12] D. Das and N. A. Touba, Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes, *Journal of Electronic Testing: Theory and Applications*, 15(1/2) 145-155 (1999).
- [13] N. Joshi, K. Wu, J. Sundararajan, and R. Karri, Concurrent Error Detection for Evolutional Functions with applications in Fault-Tolerant Cryptographic Hardware Design, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6) (2006) 1163–1169.
- [14] C. Zeng and E. J. McCluskey, Finite State Machine Synthesis with Concurrent Error Detection, *Proc. International Test Conference*, (1999) 672-679.
- [15] C. Khedhiri, M. Karmani, B. Hamdi, and K. L. Man, Concurrent Error Detection Adder Based on Two Paths Output Computation, *IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications Workshops*, (2011) 27-32.
- [16] M. Nicolaidis, On-line testing for VLSI: state of the art and trends, *Integration, the VLSI Journal*, 26(1-2) (1998) 197-209.
- [17] P. Oikonomakos and M. Zwolinski, On the Design of Self-Checking Controllers with Datapath Interactions, in *IEEE Transactions on Computers*, 55(11) (2006) 1423-1434.
- [18] Bin Talib, G.H., El-Maleh, A.H. & Sait, S.M., Design of Fault-Tolerant Adders: A Review, *Arab J Sci Eng* 43, (2018) 6667–6692.
- [19] Marc Hunger and Sybille Hellebrand, Verification and Analysis of Self-Checking Properties through ATPG, *14th IEEE International On-Line Testing Symposium*, Rhodes, Greece, (2008) 6 – 9.
- [20] P. Oikonomakos and M. Zwolinski, On the Design of Self-Checking Controllers with Data path Interactions, in *IEEE Transactions on Computers*, 55(11) (2006) 1423 – 1434.
- [21] E. Sicard, *Microwind and Dsch version 3.1*, INSA Toulouse, ISBN 2-87649-050-1, (2006).
- [22] M. Nicolaidis., Efficient implementations of self-checking adders and alus, in *23rd International Symp, On Fault-Tolerant Computing*, (1993) 586-595.
- [23] D. Rajkumar, P. K. Dutta, S. K. Sarkar., Design and implementation of 4-bit ripple carry adder using SETMOS architecture, *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, (2016) 58-61.
- [24] Akbar, M.A.; Wang, B.; Bermak, A. A High-Speed Parallel Architecture for Ripple Carry Adder with Fault Detection and Localization. *Electronics*, 10 (2021) 1791.